Displays for Statistics 5303

Lecture 17

October 11, 2002

Christopher Bingham, Instructor

612-625-7023 (St. Paul)
612-625-1024 (Minneapolis)

Class Web Page

http://www.stat.umn.edu/~kb/classes/5303

More on Factorials

Example 8.6 2×2 factorial

```
Cmd> data <- read("","exmpl8.6")
exmpl8.6       12     3
) A data set from Oehlert (2000) \emph{A First Course in Design
) and Analysis of Experiments}, New York: W. H. Freeman.
)
) Data originally from Peggy Swearigen (personal communication).
)
) Table 8.6, p. 179
) Total free amino acids in cheddar cheese aged 56 days.
) Treatments are factor/level combinations of strain r50
) not added (1) or added (2), and strain r21 not added (1)
) or added (2).  Columns are levels of r50, levels of r21,
) and amino acid response.
Read from file "TP1:Stat5303:Data:OeCh08.dat"

Cmd> data
 (1,1)           1           1       1.697
 (2,1)           2           1       2.032
 (3,1)           1           2       2.211
 (4,1)           2           2       2.091
 (5,1)           1           1       1.601
 (6,1)           2           1       2.017
 (7,1)           1           2       1.673
 (8,1)           2           2       2.255
 (9,1)           1           1        1.83
(10,1)           2           1       2.409
(11,1)           1           2       1.973
(12,1)           2           2       2.987

Cmd> makecols(data,r50, r21, y) # split up matrix

Cmd> r50 <- factor(r50); r21 <- factor(r21)

Cmd> tabs(y,r50,r21,mean:T,stddev:T,count:T)
component: mean
(1,1)      1.7093      1.9523
(2,1)      2.1527      2.4443
component: count
(1,1)           3           3
(2,1)           3           3
component: stddev
(1,1)       0.115     0.26959
(2,1)     0.22212     0.47706
```


This design is **balanced** because the sample sizes n_{ij} are the same for every combination of factor levels. Balanced designs have some advantages.

First do as g = 4 treatment experiment:

```
Cmd> treat <- makefactor(r50+10*r21);print(treat,format:"1.0f")
treat:
(1) 1 2 3 4 1 2 3 4 1 2 3 4      Factor for 4 treatments

Cmd> anova("y = treat",fstat:T)
Model used is y = treat
              DF          SS          MS           F     P-value
CONSTANT       1      51.154      51.154   563.94447  1.0526e-08
treat          3     0.87231     0.29077     3.20558    0.083373
ERROR1         8     0.72566    0.090708

Cmd> ss <- SS; df <- DF # save for later
```

A factorial ANOVA explicitly uses the factors.

```
Cmd> anova("y = r50 + r21 + r50.r21",fstat:T)
Model used is y = r50 + r21 + r50.r21
              DF          SS          MS           F     P-value
CONSTANT       1      51.154      51.154   563.94447  1.0526e-08
r50            1     0.65614     0.65614     7.23351    0.027517
r21            1      0.2144      0.2144     2.36365     0.16275
r50.r21        1   0.0017763   0.0017763     0.01958     0.89217
ERROR1         8     0.72566    0.090708
```

The model,

"y = r50 + r21 + r50.r21",

corresponds to the factorial model

$$y_{ijk} = \mu_{ijk} + \varepsilon_{ijk} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij} + \varepsilon_{ijk}$$

The ANOVA table has a line for each piece of the model.

Term	Coefficients	H_0
CONSTANT	μ	$\mu = 0$
r50	$\{\alpha_i\}$	All $\alpha_i = 0$
r21	$\{\beta_i\}$	All $\beta_j = 0$
r50.r21	$\{\alpha\beta_{ij}\}$	All $\alpha\beta_{ij} = 0$
ERROR	ε_{ijk}	

The ε_{ij} are not coefficients but they make up the error term.

When the design is *balanced*, each F-statistic tests the indicated H_0.

When the design is *not* balanced, it's more complicated. The SS and F-statistics depend on the order of the terms. F may not test what you think it does.

This is a disadvantage of lack of balance.

Some things to *know*, not just be able to look up.

With complete factorials (all factor combinations included in the experiment):

- For any factor with a levels, its main effect SS has **DF = a - 1**.
- When two factor have a and b levels, respectively, their two-way interaction SS has **DF = (a - 1)(b - 1)**
- When three factors have a, b and c levels, their three way interaction SS has DF = (a - 1)(b - 1)(c - 1)
- and so on

The main effect degrees of freedom are pretty universal.

With incomplete or fractional factorials (not all combinations in the experiment), interaction DF may be smaller, but never larger.

Suppose

- Factor A has a levels
- Factor B has b levels.

Then the two factor model for the a×b means

$$\mu_{ij} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij}$$

has more than ab parameters:

Parameters	Number
μ	1
$\{\alpha_i\}$	a
$\{\beta_j\}$	b
$\{\alpha\beta_{ij}\}$	ab
Total	ab + a + b + 1 > ab

There are many sets of μ, $\{\alpha_i\}$, $\{\beta_j\}$, and $\{\alpha\beta_{ij}\}$ with $\mu_{ij} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij}$.

Just as in the one factor case with model $\mu_i = \mu + \alpha_i$, you need restrictions to ensure unique parameter values.

The most common set of restrictions are

$$\sum_i \alpha_i = 0, \text{ so } \alpha_a = -(\alpha_1 + \alpha_2 + \ldots + \alpha_{a-1})$$

$$\sum_j \beta_j = 0, \text{ so } \beta_b = -(\beta_1 + \beta_2 + \ldots + \beta_{b-1})$$

$$\sum_i \alpha\beta_{ij} = 0, \quad j = 1,\ldots,b \quad \text{(row sums)}$$
$$\text{so } \alpha\beta_{aj} = -(\alpha\beta_{1j} + \alpha\beta_{2j} + \ldots + \alpha\beta_{a-1,j})$$

$$\sum_j \alpha\beta_{ij} = 0, \; i = 1,\ldots,a \quad \text{(column sums)}$$
$$\text{so } \alpha\beta_{ib} = -(\alpha\beta_{i1} + \alpha\beta_{i2} + \ldots + \alpha\beta_{i,b-1})$$

```
Cmd> muhat <- coefs(CONSTANT); muhat # estimate of μ
(1)      2.0647

Cmd> alphahat <- coefs(r50); alphahat # estimates of alphas
(1)    -0.23383     0.23383    Adds to zero

Cmd> betahat <- coefs(r21); betahat # estimates of betas
(1)    -0.13367     0.13367    Adds to zero

Cmd> alphabetahat <- coefs("r50.r21"); alphabetahat
(1,1)    0.012167   -0.012167   Adds to zero across rows and
(2,1)   -0.012167    0.012167   down columns
```


Each coefficient or set of coefficients contributes to all the means. The contributions can be displayed as 2 by 2 tables.

```
Cmd> term1 <- matrix(rep(muhat,4),2); term1 # from muhat
(1,1)      2.0647      2.0647  Constant term
(2,1)      2.0647      2.0647

Cmd> term2 <- hconcat(alphahat,alphahat); term2 #from alphahats
(1,1)    -0.23383    -0.23383  r50 term
(2,1)     0.23383     0.23383

Cmd> term3 <- vconcat(betahat',betahat');term3 #from betahats
(1,1)    -0.13367     0.13367  r21 term
(2,1)    -0.13367     0.13367

Cmd> term4 <- alphabetahat; term4 # from interactions
(1,1)    0.012167   -0.012167  r50 x r21 term
(2,1)   -0.012167    0.012167
```

`term1` is the same for every treatment

`term2` is constant across rows and adds to 0 down columns.

`term3` is constant down columns and adds to 0 across rows.

`term4` adds to 0 down column and across rows.

The terms for **r50** and **r21** have the form of main effect contrasts: they are constant over rows or columns.

The term for **r50.r21** (interaction or **r50** and **r21**) is like an interaction contrast: both row and column sums are 0.

You can put the terms together in several ways to give different fits to the $\overline{y}_{ij}$.

```
Cmd> tabs(y,r50,r21,mean:T) # sample treatment means
(1,1)      1.7093      1.9523
(2,1)      2.1527      2.4443

Cmd> fit_A <- term1 + term2; fit_A # Fit with just r50
(1,1)      1.8308      1.8308
(2,1)      2.2985      2.2985
```

This is a fit of the model $\mu_{ij} = \mu + \alpha_i$ in which factor B plays no role.

```
Cmd> fit_B <- term1 + term2; fit_B # fit with just r21
(1,1)      1.8308      1.8308
(2,1)      2.2985      2.2985
```

This is a fit of the model $\mu_{ij} = \mu + \beta_j$ in which factor A plays no role.


```
Cmd> fit_AplusB <- term1 + term2 + term3; fit_AplusB
(1,1)      1.6972      1.9645
(2,1)      2.1648      2.4322
```

This is a fit of the model $\mu_{ij} = \mu + \alpha_i + \beta_j$ in which the effects of A and B are *additive*. There is no interaction term.

Finally, the combination of all terms is a fit of the full model

$$\mu_{ij} = \mu + \alpha_i + \beta_j + \alpha\beta_{ij}$$

which fits the means exactly.

```
Cmd> fit_full <- term1 + term2 + term3 + term4; fit_full
(1,1)      1.7093      1.9523
(2,1)      2.1527      2.4443
```

The effects of A and B are not combined additively. There is interaction.

Estimates of parameters in a balanced design.

$$\hat{\mu} = \overline{y}_{\bullet\bullet\bullet},$$
$$\hat{\alpha}_i = \overline{y}_{i\bullet\bullet} - \hat{\mu} = \overline{y}_{i\bullet\bullet} - \overline{y}_{\bullet\bullet\bullet}$$
$$\hat{\beta}_j = \overline{y}_{\bullet j\bullet} - \hat{\mu} = \overline{y}_{\bullet j\bullet} - \overline{y}_{\bullet\bullet\bullet}$$
$$\widehat{\alpha\beta}_{ij} = \overline{y}_{ij\bullet} - \hat{\mu} - \hat{\alpha}_i - \hat{\beta}_j$$
$$= \overline{y}_{ij\bullet} - \overline{y}_{i\bullet\bullet} - \overline{y}_{\bullet j\bullet} + \overline{y}_{\bullet\bullet\bullet}$$

```
Cmd> y_dotdotdot <- tabs(y,mean:T); y_dotdotdot
(1)      2.0647

Cmd> y_idotdot <- tabs(y,r50,mean:T); y_idotdot - y_dotdotdot
(1)    -0.23383     0.23383     alphahat's

Cmd> y_dotjdot <- tabs(y,r21,mean:T);y_dotjdot - y_dotdotdot
(1)    -0.13367     0.13367     betahat's

Cmd> y_ijdot <- tabs(y,r50,r21,mean:T)

Cmd> y_ijdot - term1 - term2 - term3      alphabetahat's
(1,1)    0.012167   -0.012167
(2,1)   -0.012167    0.012167
```


Formulas for SS in the balanced case

The main effect SS are reminiscent of the single factor treatment SS

$$SS_{trt} = \sum_i n_i(\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2$$
$$= n\sum_i(\overline{y}_{i\bullet} - \overline{y}_{\bullet\bullet})^2 \text{ when } n_1 = \ldots = n_g = n$$

When there are two factors

$$SS_A = nb\sum_i(\overline{y}_{i\bullet\bullet} - \overline{y}_{\bullet\bullet\bullet})^2 = nb\sum_i\hat{\alpha}_i^2$$
$$SS_B = nb\sum_i(\overline{y}_{i\bullet\bullet} - \overline{y}_{\bullet\bullet\bullet})^2 = na\sum_j\hat{\beta}_j^2$$

and

$$SS_{AB} = n\sum_i\sum_j(\overline{y}_{ij\bullet} - \overline{y}_{i\bullet\bullet} - \overline{y}_{\bullet j\bullet} + \overline{y}_{\bullet\bullet\bullet})^2$$
$$= n\sum_i\sum_j\widehat{\alpha\beta}_{ij}^2$$

In each case, the multiplier is the number of values in the first mean in the summand, $\overline{y}_{i\bullet\bullet}$, $\overline{y}_{\bullet j\bullet}$ or $\overline{y}_{ij\bullet}$.

```
Cmd> n <- 3; a <- b <- 2

Cmd> vector(n*b*sum((y_idotdot - y_dotdotdot)^2), SS[2])
(1)     0.65614     0.65614

Cmd> vector(n*b*sum((y_dotjdot-y_dotdotdot)^2), SS[3])
(1)      0.2144      0.2144

Cmd> vector(n*sum(vector(alphabetahat^2)), SS[4])
(1)   0.0017763   0.0017763
```

I entered coefficients for the terms in a 5 by 4 factorial model.

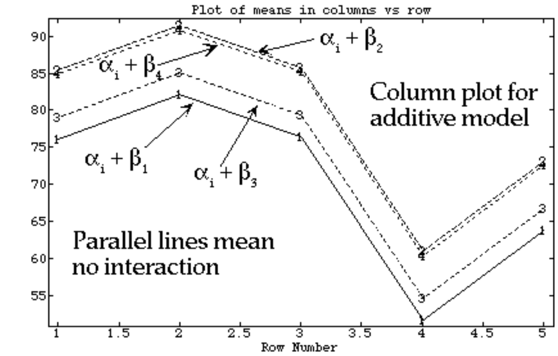
```
Cmd> print(mu,alpha,beta,alphabeta)
mu:
(1)        75.3
alpha:
(1)         6.1        12.1         6.4       -18.4        -6.2
beta:
(1)        -5.3         4.1        -2.3         3.5
alphabeta:
(1,1)         7.2        -1.8         1.4        -6.8
(2,1)         1.9        -0.9         0.4        -1.4
(3,1)        -2.6           1           3        -1.4
(4,1)        -2.7        -4.6        -6.8        14.1
(5,1)        -3.8         6.3           2        -4.5

Cmd> sum(alphabeta) # interaction sums are 0 down columns
(1,1)           0           0           0 -1.7764e-15

Cmd> sum(alphabeta') # interaction sums are 0 across rows
(1,1)           0           0           0           0           0

Cmd> mu_add <- mu + alpha + beta'; mu_add
(1,1)        76.1        85.5        79.1        84.9
(2,1)        82.1        91.5        85.1        90.9
(3,1)        76.4        85.8        79.4        85.2
(4,1)        51.6          61        54.6        60.4
(5,1)        63.8        73.2        66.8        72.6
```

MacAnova comments.

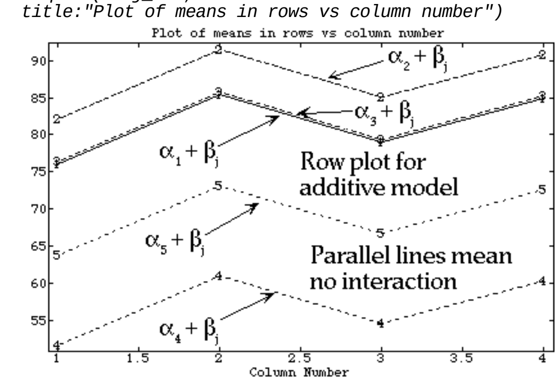
`alphabeta'` is the transpose of `alphabeta` with rows swapped with columns.

`mu + alpha + beta'` combines a scalar (`mu`), a column vector (`alpha`) and a row vector (`beta'`, the transpose of `beta`) to make a table.

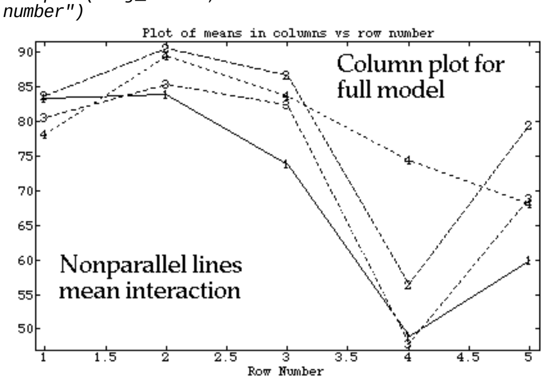

```
Cmd> colplot(muij_add, title:"Plot of means in columns vs row number")
```




```
Cmd> rowplot(muij_add,\
     title:"Plot of means in rows vs column number")
```




```
Cmd> muij_nonadd <- muij + alphabeta # full model

Cmd> colplot(muij_nonadd, title:"Plot of means in columns vs
row number")
```




```
Cmd> rowplot(muij_nonadd, title:"Plot of means in rows vs
column number")
```